

# 2D Sparse Signal Recovery via 2D Orthogonal Matching Pursuit

Yong Fang, Bormin Huang, and Jiaji Wu

## Abstract

Recovery algorithms play a key role in compressive sampling (CS). Most of current CS recovery algorithms are originally designed for one-dimensional (1D) signal, while many practical signals are two-dimensional (2D). By utilizing 2D separable sampling, 2D signal recovery problem can be converted into 1D signal recovery problem so that ordinary 1D recovery algorithms, e.g. orthogonal matching pursuit (OMP), can be applied directly. However, even with 2D separable sampling, the memory usage and complexity at the decoder is still high. This paper develops a novel recovery algorithm called 2D-OMP, which is an extension of 1D-OMP. In the 2D-OMP, each atom in the dictionary is a matrix. At each iteration, the decoder projects the sample matrix onto 2D atoms to select the best matched atom, and then renews the weights for all the already selected atoms via the least squares. We show that 2D-OMP is in fact equivalent to 1D-OMP, but it reduces recovery complexity and memory usage significantly. What's more important, by utilizing the same methodology used in this paper, one can even obtain higher dimensional OMP (say 3D-OMP, etc.) with ease.

## Index Terms

Compressive Sampling, 2D Sparse Signal, Recovery Algorithm, Orthogonal Matching Pursuit.

This research was supported by National Science Foundation of China (NSFC) (grant no. 61001100) and Provincial Science Foundation of Shaanxi, China (grant no. 2010K06-15).

Y. Fang is with College of Information Engineering, Northwest A&F University, Shaanxi Yangling 712100, China (email: yfang79@gmail.com).

B. Huang is with Space Science and Engineering Center, University of Wisconsin-Madison, 1225 W. Dayton St. Madison, WI 53706, USA (email: bormin@ssec.wisc.edu).

J. Wu is with Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, Xidian University, Xi'an 710071, China (email: wujj@mail.xidian.edu.cn).

## I. INTRODUCTION

Let  $\mathbf{x} \in \mathbb{R}^n$  be a one-dimensional (1D) signal and  $\Psi \in \mathbb{R}^{n \times n}$  an orthonormal transform matrix, where  $\mathbb{R}$  is the set of real numbers. If  $\mathbf{x} = \Psi \mathbf{z}$  and there are only  $k \ll n$  spikes (nonzero entries) in  $\mathbf{z}$ , we say that  $\mathbf{x}$  is  $k$ -sparse in  $\Psi$  domain. We sample  $\mathbf{x}$  by  $\Phi \in \mathbb{R}^{m \times n}$  ( $k < m < n$ ) to get  $\mathbf{y} = \Phi \mathbf{x} = \mathbf{A} \mathbf{z} \in \mathbb{R}^m$ , where  $\mathbf{A} = \Phi \Psi$ . If  $\Phi$  obeys the order- $k$  restricted isometry property (RIP) and has low coherence with  $\Psi$ , then  $\mathbf{z}$  (and in turn  $\mathbf{x}$ ) can be effectively recovered from  $\mathbf{y}$  [1], [2], [3]. Many algorithms have been proposed to recover  $\mathbf{x}$  from its random sample  $\mathbf{y}$ , e.g. linear programming (LP) [4] and orthogonal matching pursuit (OMP) [5]. For a detailed overview on recovery algorithms, please refer to [6].

In practice, many signals, e.g. image, video, etc, are two-dimensional (2D). A straightforward implementation of 2D compressive sampling (CS) is to stretch 2D matrices into 1D vectors. However, such direct stretching increases exponentially the complexity and memory usage at both encoder and decoder. An alternative to 1D stretching is to sample rows and columns of 2D signals independently by using separable operators [7]. Through 2D separable sampling, encoding complexity is exponentially reduced. However, as the recovery problem is converted into a standard 1D  $\ell_1$ -minimization problem, decoding complexity is still very high.

As a representative sparse signal recovery algorithm, the OMP achieves good performance with low complexity. The OMP is originally designed for 1D signal recovery. To reduce the complexity of 2D signal recovery, this paper extends the 1D-OMP to obtain the 2D-OMP. We prove that with 2D separable sampling, 2D-OMP is in fact equivalent to 1D-OMP, so that both algorithms will output exactly the same results. However, the complexity and memory usage of 2D-OMP is much lower than that of 1D-OMP. Thus, 2D-OMP can be used as an alternative to 1D-OMP in 2D sparse signal recovery.

This paper is arranged as follows. Section II first briefly reviews the principles of 2D separable sampling and 1D-OMP, and then makes a detailed analysis on the complexity of 1D-OMP. In Section III, we deduce the 2D-OMP algorithm, reveal the equivalence of 2D-OMP to 1D-OMP, and compare the complexity and memory usage of 2D-OMP with that of 1D-OMP. In Section IV, simulation results are reported. Finally, Section V concludes this paper.

## II. 1D ORTHOGONAL MATCHING PURSUIT WITH 2D SEPARABLE SAMPLING

### A. 2D Separable Sampling

The principle of 2D separable sampling [7] is as follows. Let  $\mathbf{X} \in \mathbb{R}^{n \times n}$  be a 2D signal which is  $k$ -sparse in  $\Psi$  domain, i.e.  $\mathbf{X} = \Psi \mathbf{Z} \Psi^T$  and there are only  $k \ll n^2$  spikes in  $\mathbf{Z}$ , where  $(\cdot)^T$  denotes the transpose. For simplicity, we use the same operator  $\Phi$  to sample the rows and columns of  $\mathbf{X}$  independently

to get  $\mathbf{Y} = \Phi \mathbf{X} \Phi^T = \mathbf{A} \mathbf{Z} \mathbf{A}^T \in \mathbb{R}^{m \times m}$ . Let  $\mathbf{y} \in \mathbb{R}^{m^2}$  be the 1D stretched vector of  $\mathbf{Y}$  and  $\mathbf{z} \in \mathbb{R}^{n^2}$  the 1D stretched vector of  $\mathbf{Z}$ . It was proved that

$$\mathbf{y} = (\Phi \otimes \Phi)(\Psi \otimes \Psi)\mathbf{z} = \Omega \mathbf{z}, \quad (1)$$

where  $\otimes$  denotes the Kronecker product [7]. It is easy to prove  $(\Phi \otimes \Phi)(\Psi \otimes \Psi) = (\Phi \Psi) \otimes (\Phi \Psi)$ , hence  $\Omega = \mathbf{A} \otimes \mathbf{A}$ . Now this is just a standard 1D sparse signal recovery problem which can be attacked by the OMP.

### B. 1D Orthogonal Matching Pursuit

Let  $\Omega = (\omega_1, \dots, \omega_{n^2})$ , where  $\omega_i \in \mathbb{R}^{m^2}$  is the  $i$ -th column of  $\Omega$ . We call  $\Omega$  the *dictionary* and  $\omega_i$  an *atom*. The main idea of 1D-OMP is to represent  $\mathbf{y}$  as a weighted sum of as few atoms as possible.

Algorithm 1 gives main steps of 1D-OMP. To implement 1D-OMP, we need two auxiliary variables. First, to avoid atom reselection, set  $\mathbf{i}$  is defined to record the indices of those atoms that are allowed to be selected in the future (excluding those already selected atoms). Second, vector  $\mathbf{r} \in \mathbb{R}^{m^2}$  is defined to hold the residual after removing the selected atoms from  $\mathbf{y}$ . Initially,  $\mathbf{r}$  is set to  $\mathbf{y}$ . Then at each iteration, the decoder picks from the dictionary the atom that best matches the residual and then renews the weights for all the already selected atoms via the least squares.

### C. Complexity Analysis

We decompose the iteration of 1D-OMP into the following steps and analyze its complexity step by step.

1) *Project*: The projection of residual  $\mathbf{r}$  onto atom  $\omega_{i'}$  is  $\langle \mathbf{r}, \omega_{i'} \rangle / \|\omega_{i'}\|_2$ , where  $\langle \cdot, \cdot \rangle$  denotes the inner product between two vectors and  $\|\cdot\|_2$  denotes the  $\ell_2$ -norm of a vector. Let  $\boldsymbol{\rho} = (\|\omega_1\|_2, \dots, \|\omega_{n^2}\|_2)^T$ , then this step can be implemented by  $\Omega^T \mathbf{r} ./ \boldsymbol{\rho}$ , where  $./$  denotes dot division. The complexity of this step is dominated by  $(n^2 \times m^2) \times (m^2 \times 1)$  matrix-vector multiplication. Hence the complexity of this step is  $O(m^2 n^2)$ .

2) *Select Best Matched Unselected Atom*: This step selects from unselected atoms the atom with the maximal absolute value of projection. As there are  $n^2$  atoms and  $k \ll n^2$ , the complexity of this step is approximately  $O(n^2)$ , negligible compared with the *Project* step.

3) *Renew Weights*: Let  $\Omega_i = (\omega_{i_1}, \dots, \omega_{i_t})$ , then  $\sum_{t'=1}^t u_{t'} \omega_{i_{t'}} = \Omega_i \mathbf{u}$ . According to linear algebra,

$$\arg \min_{\mathbf{u}} \|\mathbf{y} - \Omega_i \mathbf{u}\|_2 = \mathbf{Q}^{-1} \mathbf{g}, \quad (2)$$

where  $\mathbf{Q} = \Omega_i^T \Omega_i \in \mathbb{R}^{t \times t}$  and  $\mathbf{g} = \Omega_i^T \mathbf{y} \in \mathbb{R}^t$ . The complexity to calculate  $\mathbf{Q}$  and  $\mathbf{g}$  depends on  $t$ . For  $t \leq k \ll n^2$ , the complexity of this step is negligible compared with the *Project* step.

---

**Algorithm 1:** 1D Orthogonal Matching Pursuit with 2D Separable Sampling

---

**Input:**

- $\Omega \in \mathbb{R}^{m^2 \times n^2}$ : sampling matrix
- $\mathbf{y} \in \mathbb{R}^{m^2}$ : sample
- $k$ : sparsity level

**Output:**

- $\tilde{\mathbf{z}} \in \mathbb{R}^{n^2}$ : reconstruction of the ideal signal  $\mathbf{z}$

**Auxiliary Variables:**

- $\mathbf{r} \in \mathbb{R}^{m^2}$ : residual
- $\mathbf{i}$ : set of the indices of atoms that are allowed to be selected in the future

**Initialization:**

- $\mathbf{r} \leftarrow \mathbf{y}$
- $\mathbf{i} \leftarrow \{1, 2, \dots, n^2\}$

**for**  $t \leftarrow 1$  **to**  $k$  **do**

$$\left[ \begin{array}{l} i_t \leftarrow \arg \max_{i' \in \mathbf{i}} \frac{|\langle \mathbf{r}, \boldsymbol{\omega}_{i'} \rangle|}{\|\boldsymbol{\omega}_{i'}\|_2}; \\ \mathbf{i} \leftarrow \mathbf{i} \setminus i_t; \\ \hat{\mathbf{u}} \leftarrow \arg \min_{\mathbf{u}} \left\| \mathbf{y} - \sum_{t'=1}^t u_{t'} \boldsymbol{\omega}_{i_{t'}} \right\|_2; \\ \mathbf{r} \leftarrow \mathbf{y} - \sum_{t'=1}^t \hat{u}_{t'} \boldsymbol{\omega}_{i_{t'}}; \end{array} \right.$$

**for**  $t \leftarrow 1$  **to**  $k$  **do**

$$\left[ \begin{array}{l} \tilde{z}_{i_t} \leftarrow \hat{u}_t; \end{array} \right.$$


---

4) *Update Residual*: The complexity of this step depends on  $t$ . For  $t \leq k \ll n^2$ , the complexity of this step is negligible compared with the *Project* step.

Based on the above analysis, we conclude that the complexity of 1D-OMP is dominated by the *Project* step and its complexity is  $O(m^2 n^2)$ .

### III. 2D ORTHOGONAL MATCHING PURSUIT

This section develops the 2D-OMP algorithm whose main idea is to represent 2D signal  $\mathbf{Y}$  as a weighted sum of 2D atoms that are selected from an over-complete dictionary. We first redefine the concepts of atom, dictionary, and projection for 2D signals. Then we give the 2D-OMP algorithm. We reveal the

equivalence of 2D-OMP to 1D-OMP and compare the complexity and memory usage of 2D-OMP with that of 1D-OMP.

#### A. Definition

Let  $\mathbf{X} \in \mathbb{R}^{n \times n}$  be a 2D signal that is  $k$ -sparse in  $\Psi$  domain and  $\mathbf{Y} = \Phi \mathbf{X} \Phi^T = \mathbf{A} \mathbf{Z} \mathbf{A}^T \in \mathbb{R}^{m \times m}$ . Let  $\mathbf{A} = (\mathbf{a}_1, \dots, \mathbf{a}_n)$ , where  $\mathbf{a}_i$  is the  $i$ -th column of  $\mathbf{A}$ . We redefine dictionary, atom, and projection as follows.

1) *Dictionary and Atom*: In the 2D-OMP, the dictionary contains  $n^2$  atoms and each atom is an  $m \times m$  matrix. Let  $\mathbf{B}_{i,j}$  be the  $(i, j)$ -th atom, then  $\mathbf{B}_{i,j}$  is the outer product of  $\mathbf{a}_i$  and  $\mathbf{a}_j$

$$\mathbf{B}_{i,j} = \mathbf{a}_i \mathbf{a}_j^T = \begin{pmatrix} a_{1i}a_{1j} & \cdots & a_{1i}a_{mj} \\ \vdots & \ddots & \vdots \\ a_{mi}a_{1j} & \cdots & a_{mi}a_{mj} \end{pmatrix}. \quad (3)$$

Now  $\mathbf{Y}$  can be represented by the weighted sum of  $\mathbf{B}_{i,j}$ , i.e.

$$\mathbf{Y} = \sum_{i=1}^n \sum_{j=1}^n z_{i,j} \mathbf{B}_{i,j}. \quad (4)$$

2) *Projection*: The projection of  $\mathbf{Y}$  onto  $\mathbf{B}_{i,j}$  is

$$\frac{\langle \mathbf{Y}, \mathbf{B}_{i,j} \rangle}{\|\mathbf{B}_{i,j}\|_2}, \quad (5)$$

where  $\langle \mathbf{Y}, \mathbf{B}_{i,j} \rangle \triangleq \mathbf{a}_i^T \mathbf{Y} \mathbf{a}_j$  and  $\|\mathbf{B}_{i,j}\|_2$  is the Frobenius norm of  $\mathbf{B}_{i,j}$ , i.e.

$$\|\mathbf{B}_{i,j}\|_2 \triangleq \sqrt{\sum_{i'=1}^m \sum_{j'=1}^m (a_{i'i}a_{j'j})^2} = \|\mathbf{a}_i\|_2 \|\mathbf{a}_j\|_2. \quad (6)$$

#### B. Algorithm Description

Algorithm 2 gives main steps of 2D-OMP. To implement the 2D-OMP algorithm, we also need two auxiliary variables. First, to avoid atom reselection, set  $(\mathbf{i}, \mathbf{j})$  is defined to record the coordinates of those atoms that are allowed to be selected in the future (excluding those already selected atoms), where  $\mathbf{i}$  for row indices and  $\mathbf{j}$  for column indices. Second,  $\mathbf{R} \in \mathbb{R}^{m \times m}$  is defined to hold the residual after removing the selected atoms from  $\mathbf{Y}$ . Initially,  $\mathbf{R}$  is set to  $\mathbf{Y}$ . Then at each iteration, the decoder first searches for the best matched atom in the dictionary and then renews the weights for all the already selected atoms via the least squares.

---

**Algorithm 2:** 2D Orthogonal Matching Pursuit

---

**Input:**

- $\mathbf{A} \in \mathbb{R}^{m \times n}$ : sampling matrix
- $\mathbf{Y} \in \mathbb{R}^{m \times m}$ : sample
- $k$ : sparsity level

**Output:**

- $\tilde{\mathbf{Z}} \in \mathbb{R}^{n \times n}$ : reconstruction of the ideal signal  $\mathbf{Z}$

**Variable:**

- $\mathbf{R} \in \mathbb{R}^{m \times m}$ : residual
- $(i, j)$ : set of the coordinates of atoms that are allowed to be selected in the future,  $i$  for row indices and  $j$  for column indices

**Initialization:**

- $\mathbf{R} \leftarrow \mathbf{Y}$
- $(i, j) \leftarrow \{(1, 1), (1, 2), \dots, (n, n)\}$

**for**  $t \leftarrow 1$  **to**  $k$  **do**

$$\left| \begin{array}{l} (i_t, j_t) \leftarrow \arg \max_{(i', j') \in (i, j)} \frac{|\langle \mathbf{R}, \mathbf{B}_{i', j'} \rangle|}{\|\mathbf{B}_{i', j'}\|_2}; \\ (i, j) \leftarrow (i, j) \setminus (i_t, j_t); \\ \hat{\mathbf{u}} \leftarrow \arg \min_{\mathbf{u}} \left\| \mathbf{Y} - \sum_{t'=1}^t u_{t'} \mathbf{B}_{i_{t'}, j_{t'}} \right\|_2; \\ \mathbf{R} \leftarrow \mathbf{Y} - \sum_{t'=1}^t \hat{u}_{t'} \mathbf{B}_{i_{t'}, j_{t'}}; \end{array} \right.$$

**for**  $t \leftarrow 1$  **to**  $k$  **do**

$$\left| \begin{array}{l} \tilde{z}_{i_t, j_t} \leftarrow \hat{u}_t; \end{array} \right.$$


---

1) *Least Squares*: The weighted sum of  $t$  selected atoms constructs an approximation to  $\mathbf{Y}$ . Let

$$\mathbf{R} = \mathbf{Y} - \sum_{t'=1}^t u_{i_{t'}, j_{t'}} \mathbf{B}_{i_{t'}, j_{t'}}. \quad (7)$$

We model the problem as finding the optimal  $\mathbf{u} = (u_{i_1, j_1}, \dots, u_{i_t, j_t})^T$  that minimizes the Frobenius norm of  $\mathbf{R}$ , which is in fact equivalent to the least squares problem. As  $\|\mathbf{R}\|_2^2 = \text{tr}(\mathbf{R}\mathbf{R}^T)$ , where  $\text{tr}(\cdot)$  is the trace of a matrix, the problem is equivalent to

$$\hat{\mathbf{u}} = \arg \min_{\mathbf{u}} \text{tr}(\mathbf{R}\mathbf{R}^T). \quad (8)$$

Using (7) and  $\text{tr}((\cdot)^T) = \text{tr}(\cdot)$ , we have

$$\begin{aligned} \text{tr}(\mathbf{R}\mathbf{R}^T) &= \text{tr}(\mathbf{Y}\mathbf{Y}^T) - 2 \sum_{t'=1}^t u_{i_{t'}, j_{t'}} \text{tr}(\mathbf{Y}\mathbf{B}_{i_{t'}, j_{t'}}^T) + \\ &\quad \sum_{t'=1}^t \sum_{s'=1}^t u_{i_{t'}, j_{t'}} u_{i_{s'}, j_{s'}} \text{tr}(\mathbf{B}_{i_{t'}, j_{t'}} \mathbf{B}_{i_{s'}, j_{s'}}^T) \\ &= \|\mathbf{Y}\|_2^2 + \mathbf{u}^T \mathbf{H} \mathbf{u} - 2 \mathbf{f}^T \mathbf{u}, \end{aligned} \quad (9)$$

where

$$\mathbf{H} = \begin{pmatrix} \text{tr}(\mathbf{B}_{i_1, j_1} \mathbf{B}_{i_1, j_1}^T) & \cdots & \text{tr}(\mathbf{B}_{i_1, j_1} \mathbf{B}_{i_t, j_t}^T) \\ \vdots & \ddots & \vdots \\ \text{tr}(\mathbf{B}_{i_t, j_t} \mathbf{B}_{i_1, j_1}^T) & \cdots & \text{tr}(\mathbf{B}_{i_t, j_t} \mathbf{B}_{i_t, j_t}^T) \end{pmatrix} \quad (10)$$

and

$$\mathbf{f} = (\text{tr}(\mathbf{Y}\mathbf{B}_{i_1, j_1}^T), \dots, \text{tr}(\mathbf{Y}\mathbf{B}_{i_t, j_t}^T))^T. \quad (11)$$

When  $\text{tr}(\mathbf{R}\mathbf{R}^T)$  takes the minimum, there must be

$$\frac{\partial \text{tr}(\mathbf{R}\mathbf{R}^T)}{\partial \mathbf{u}} = 2\mathbf{H}\mathbf{u} - 2\mathbf{f} = \mathbf{0}. \quad (12)$$

Hence

$$\hat{\mathbf{u}} = \mathbf{H}^{-1} \mathbf{f}. \quad (13)$$

2) *Calculation of  $\mathbf{H}$  and  $\mathbf{f}$* : It is easy to get

$$\begin{aligned} \text{tr}(\mathbf{B}_{i_{t'}, j_{t'}} \mathbf{B}_{i_{s'}, j_{s'}}^T) &= \text{tr}(\mathbf{a}_{i_{t'}} \mathbf{a}_{j_{t'}}^T \mathbf{a}_{j_{s'}} \mathbf{a}_{i_{s'}}^T) \\ &= \langle \mathbf{a}_{i_{t'}}, \mathbf{a}_{i_{s'}} \rangle \langle \mathbf{a}_{j_{t'}}, \mathbf{a}_{j_{s'}} \rangle. \end{aligned} \quad (14)$$

Let  $\langle \mathbf{B}_{i_{t'}, j_{t'}}, \mathbf{B}_{i_{s'}, j_{s'}} \rangle \triangleq \langle \mathbf{a}_{i_{t'}}, \mathbf{a}_{i_{s'}} \rangle \langle \mathbf{a}_{j_{t'}}, \mathbf{a}_{j_{s'}} \rangle$ , then

$$\mathbf{H} = \begin{pmatrix} \langle \mathbf{B}_{i_1, j_1}, \mathbf{B}_{i_1, j_1} \rangle & \cdots & \langle \mathbf{B}_{i_1, j_1}, \mathbf{B}_{i_t, j_t} \rangle \\ \vdots & \ddots & \vdots \\ \langle \mathbf{B}_{i_t, j_t}, \mathbf{B}_{i_1, j_1} \rangle & \cdots & \langle \mathbf{B}_{i_t, j_t}, \mathbf{B}_{i_t, j_t} \rangle \end{pmatrix}. \quad (15)$$

Similarly, for  $\mathbf{f}$ , because

$$\text{tr}(\mathbf{Y}\mathbf{B}_{i, j}^T) = \text{tr}(\mathbf{Y} \mathbf{a}_j \mathbf{a}_i^T) = \mathbf{a}_i^T \mathbf{Y} \mathbf{a}_j = \langle \mathbf{Y}, \mathbf{B}_{i, j} \rangle, \quad (16)$$

we have

$$\mathbf{f} = (\langle \mathbf{Y}, \mathbf{B}_{i_1, j_1} \rangle, \dots, \langle \mathbf{Y}, \mathbf{B}_{i_t, j_t} \rangle)^T. \quad (17)$$

### C. Equivalence of 2D-OMP to 1D-OMP

Obviously, the  $(n(i-1) + j)$ -th atom of  $\mathbf{\Omega}$  is

$$\boldsymbol{\omega}_{n(i-1)+j} = \begin{pmatrix} a_{1i}\mathbf{a}_j \\ \vdots \\ a_{mi}\mathbf{a}_j \end{pmatrix}. \quad (18)$$

Compared with (3), it can be found that  $\boldsymbol{\omega}_{n(i-1)+j}$  is just the 1D stretched vector of  $\mathbf{B}_{i,j}$ . Hence, the Frobenius norm of  $\mathbf{B}_{i,j}$  will equal the  $\ell_2$ -norm of  $\boldsymbol{\omega}_{n(i-1)+j}$ .

Then we prove that the projection of  $\mathbf{R}$  onto  $\mathbf{B}_{i,j}$  equals the projection of  $\mathbf{r}$  onto  $\boldsymbol{\omega}_{n(i-1)+j}$ . Obviously,

$$\begin{aligned} \langle \mathbf{r}, \boldsymbol{\omega}_{n(i-1)+j} \rangle &= \sum_{i'=1}^m \sum_{j'=1}^m a_{i'i} a_{j'j} \mathbf{r}_{m(i'-1)+j'} \\ &= \mathbf{a}_i^T \mathbf{R} \mathbf{a}_j = \langle \mathbf{R}, \mathbf{B}_{i,j} \rangle. \end{aligned} \quad (19)$$

Hence

$$\frac{|\langle \mathbf{r}, \boldsymbol{\omega}_{n(i-1)+j} \rangle|}{\|\boldsymbol{\omega}_{n(i-1)+j}\|_2} = \frac{|\langle \mathbf{R}, \mathbf{B}_{i,j} \rangle|}{\|\mathbf{B}_{i,j}\|_2}. \quad (20)$$

It means: at each iteration of 1D-OMP and 2D-OMP, the same atom will be selected.

Finally, as  $\mathbf{r}$  is the 1D stretched vector of  $\mathbf{R}$ ,  $\|\mathbf{R}\|_2 = \|\mathbf{r}\|_2$ . Hence, the least squares in 1D-OMP and 2D-OMP will output exactly the same results (in fact, it can be easily proved that  $\mathbf{Q} = \mathbf{H}$  and  $\mathbf{f} = \mathbf{g}$ ).

Based on the above analysis, we draw the conclusion that 2D-OMP is equivalent to 1D-OMP.

### D. Complexity Analysis

Below we analyze the complexity of 2D-OMP step by step.

1) *Project*: Let  $\mathbf{P}$  be an  $n \times n$  matrix whose  $(i, j)$ -th element is  $\|\mathbf{B}_{i,j}\|_2$ . Then this step can be implemented by  $\mathbf{A}^T \mathbf{R} \mathbf{A} / \mathbf{P}$ . The complexity of this step is dominated by  $(n \times m) \times (m \times m)$  matrix-matrix multiplication and  $(n \times m) \times (m \times n)$  matrix-matrix multiplication. As  $m < n$ , the complexity of this step is  $O(mn^2)$ .

2) *Select Best Matched Unselected Atom*: Since there are  $n^2$  atoms and  $k \ll n^2$ , the complexity of this step is approximately  $O(n^2)$ , negligible compared with the *Project* step.

3) *Renew Weights*: From (14), it can be seen that the complexity to calculate  $\langle \mathbf{B}_{i_{t'}, j_{t'}}, \mathbf{B}_{i_{s'}, j_{s'}} \rangle$  is  $O(m)$ . Since  $\langle \mathbf{Y}, \mathbf{B}_{i,j} \rangle = \mathbf{a}_i^T \mathbf{Y} \mathbf{a}_j$ , the complexity to calculate  $\langle \mathbf{Y}, \mathbf{B}_{i,j} \rangle$  is  $O(m^2)$ . The complexity to calculate  $\mathbf{H}$  and  $\mathbf{f}$  depends on  $t$ . For  $t \leq k \ll n^2$ , the complexity of this step is negligible compared with the *Project* step.



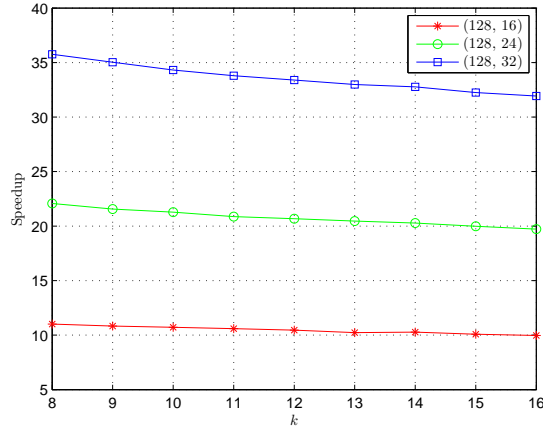


Fig. 1. Speedup of 2D-OMP over 1D-OMP.

4) *Update Residual*: The complexity of this step depends on  $t$ . For  $t \leq k \ll n^2$ , the complexity of this step is negligible compared with the *Project* step.

Based on the above analysis, we draw the conclusion that the complexity of 2D-OMP is  $O(mn^2)$ , roughly  $1/m$  of that of 1D-OMP.

#### E. Memory Usage Analysis

For 1D-OMP, an  $m^2 \times n^2$  matrix is needed to hold  $\mathbf{\Omega}$ , so the memory usage is  $O(m^2n^2)$ , while for 2D-OMP,  $\mathbf{\Omega}$  is replaced by  $\mathbf{A}$ , so the memory usage is reduced to  $O(mn)$ .

### IV. EXPERIMENTAL RESULTS

We have written both 2D-OMP and 1D-OMP algorithms in MATLAB [8]. We present herein the results under three typical settings, i.e.  $(n, m) = (128, 16)$ ,  $(128, 24)$ , and  $(128, 32)$ . For each setting, we increase sparsity level  $k$  from 8 to 16. The transform matrix  $\mathbf{\Psi}$  is  $128 \times 128$  2D discrete cosine transform (DCT) matrix. The sensing matrix  $\mathbf{\Phi}$  is formed by sampling independent and identically-distributed (i.i.d.) entries from standard Gaussian distribution by using the *randn* function. 2D Sparse signal is obtained by using the *sprandn* function with density  $k/n^2$ .

We run the MATLAB codes on Intel(R) Core(TM) i7 CPU with 12GB memory and collect the total running time of  $10^3$  trials for 1D-OMP and 2D-OMP respectively. Because two algorithms output exactly the same results, only the speedup of 2D-OMP over 1D-OMP with respect to  $k$  is reported in Fig. 1. From Fig. 1, we can draw two conclusions:

- 1) As  $k$  increases, the speedup of 2D-OMP over 1D-OMP descends gradually. This is because for small  $k$ , the complexity of 2D-OMP and 1D-OMP is dominated by the *Project* step. As  $k$  increases, the complexity of other steps will weight heavier. Especially, at the *Renew Weights* step, the complexity of  $t \times t$  matrix inverse is  $O(t^3)$ , which will ascend quickly as  $t$  increases.
- 2) As  $m$  increases, the speedup of 2D-OMP over 1D-OMP becomes more significant. When  $m = 16$ , the speedup of 2D-OMP over 1D-OMP ranges from 10 to 11 times, while when  $m = 32$ , the speedup of 2D-OMP over 1D-OMP ranges from 32 to 35 times. This is because the speedup of 2D-OMP over 1D-OMP comes mainly from the *Project* step, while at other steps 2D-OMP shows little superiority over the 1D-OMP. As  $m$  increases, the complexity of the *Project* step weights heavier, which explains the above phenomenon.

## V. CONCLUSION

For 2D sparse signal recovery, this paper develops 2D-OMP algorithm. We prove that 2D-OMP is equivalent to 1D-OMP, but it reduces recovery complexity and memory usage. Hence, 2D-OMP can be used as an alternative to 1D-OMP in such scenarios as compressive imaging, image compression, etc.

Following the deduction in this paper, the extension of 2D-OMP to higher dimensional OMP is straightforward. For example, by utilizing 3D separable sampling, 3D-OMP can be obtained by defining each atom as a 3D matrix. Then at each iteration, the decoder projects 3D sample matrix onto 3D atoms to select the best matched atom, and then renews the weights for all the already selected atoms via the least squares. 3D-OMP can find its use in hyperspectral image compression.

## REFERENCES

- [1] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inform. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.
- [2] E. Candes and T. Tao, "Near optimal signal recovery from random projections: Universal encoding strategies?," *IEEE Trans. Inform. Theory*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.
- [3] D. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.
- [4] E. Candes and T. Tao, "Decoding by linear programming," *IEEE Trans. Inform. Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.
- [5] J. Tropp and A. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inform. Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.
- [6] G. Pope, "Compressive sensing: A summary of reconstruction algorithms," *Master Thesis*, Feb. 2009.
- [7] Y. Rivenson and A. Stern, "Compressed imaging with a separable sensing operator," *IEEE Signal Process. Lett.*, vol. 16, no. 6, pp. 449–452, Jun. 2009.
- [8] <http://www.wavesharp.com/fangyong/>